

Remarks

Reconsideration of the application and allowance of all pending claims are respectfully requested. Claims 1-69 remain pending. Applicants respectfully request careful consideration of the remarks below, which address new grounds of rejection.

In the Final Office Action dated February 23, 2004, claims 1-3, 9, 22-24, 30, 43-45, 49-50 and 57 are rejected under 35 U.S.C. 102(b) as being anticipated by Lynch (U.S. Patent No. 5,829,031); claims 4-5, 25-26 and 52-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Kahle (U.S. Patent No. 6,574,712); claims 6-7, 27-28 and 54-55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Kahle and further in view of Ryan (U.S. Patent No. 5,367,656); claims 8, 29 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Lopez-Aguado et al. (U.S. Patent No. 6,317,810); claims 10, 31 and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Ryan; claims 11, 32 and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Ryan and further in view of Lopez-Aguado; claims 12, 15-16, 33, 36-37, 45-46, 60 and 63-64 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Sollars (U.S. Patent No. 5,799,164); claims 13, 34 and 61 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch and Sollars and further in view of Ryan; claims 14, 35 and 62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch and Sollars and further in view of Lopez-Aguado; claims 17, 20, 38, 41, 47-48, 65 and 68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Mason, Jr. (U.S. Patent No. 5,884,098); claims 18, 39 and 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Mason and further in view of Ryan; claims 19, 40 and 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Mason and further in view of Lopez-Aguado et al.; and claims 21, 42 and 69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lynch in view of Mason and further in view of Ryan. Applicants respectfully, but most strenuously, traverse these rejections for the reasons below.

In one aspect of applicants' invention, applicants claim a method of managing the prefetching of data files (e.g., independent claim 1). The method includes, for instance, detecting a pattern of requests for data of multiple files, wherein the pattern is based on one or more user-

defined attributes of the multiple files; and prefetching data of a plurality of files, in response to the detecting indicating the pattern. Thus, in this aspect of applicants' claimed invention, the pattern being detected is based on one or more user-defined attributes of the multiple files, i.e., attributes of the files defined by the user. This is very different from the teachings of Lynch.

Lynch describes detecting a pattern of instructions that is indicative of a specific function. Lynch does not describe detecting a pattern that is based on file attributes defined by the user. File attributes are characteristics of the files (see, e.g., definition of attributes in Webster's Ninth New Collegiate Dictionary), such as file name, the directory to contain the files, etc. As claimed, these attributes are provided by the user. User defined attributes are just that attributes defined by the user – and in this case, those attributes defined by the user are related to files that include data. This is very different from the teachings of Lynch.

In Lynch, a pattern is detected, but the pattern is not based on file attributes defined by a user. In fact, the pattern in Lynch is not based on attributes of a file at all, but instead, on occurrence of particular instructions which are indicative of a specific function (Col. 4, lines 7-10; Col. 8, line 55; Col. 10, lines 1-6). Lynch does not even mention files, much less attributes of those files in its detecting. The only mention of a file in Lynch is with respect to a register file 34 from which operands are retrieved. The register file nor any other file is described with reference to detecting a pattern. Since files are not mentioned, it follows that attributes of those files, and in particular, user defined attributes of those files, are not described, either. Thus, there is no description, teaching or suggestion in Lynch of detecting a pattern that is based on one or more user-defined attributes of the files.

The microprocessor used to detect patterns in Lynch does not use and has no knowledge of user-defined attributes of files. Instead, Lynch is limited to detecting a pattern based on analyzing address registers of instructions. At Col. 8, line 66 Col. 9, line 9, Lynch describes, with reference to FIG. 3 thereof, the group of instructions that indicate the pattern of data accesses:

Heuristic processing unit 36 begins detection of the group of instructions at step 50 by capturing a register used as an addressing register (i.e., a register storing a value used to form the address of a memory operation). Subsequent instructions are

analyzed by heuristic processing unit 36 according to steps 52, 54, and 56. If the register is found to be the target of another instruction (step 52), and the instruction modifies the register by a constant value (step 54), and the register is used in a subsequent instruction to form an address (step 56), then heuristic processing unit 36 notifies data cache 24 to begin prefetching (step 58).

Thus, the data access pattern detected in Lynch is based on analyzing registers of instructions and the formation of a memory address. This is very different from this aspect of applicants' claimed invention in which the pattern of data requests that is being detected is based on one or more attributes of multiple files that are defined by the user.

Since Lynch fails to describe, teach or suggest files in its detecting, attributes of files or detecting a pattern that is based on user-defined attributes, applicants respectfully submit that Lynch does not anticipate applicants' claimed invention. Thus, applicants respectfully request withdrawal of the §102 rejection of claim 1, similar independent claims, and all claims that depend therefrom.

In a further aspect of applicants' invention, applicants claim a method of managing the prefetching of data (e.g., independent claim 12). The method includes, for instance, controlling, subsequent to determining that prefetching of data is to occur, a rate at which data of a plurality of files is prefetched by pacing the prefetching based upon requests for data; and prefetching the data of the plurality of files, in response to the controlling. Thus, in this aspect of applicants' claimed invention, after it is determined that prefetching of data is to occur, a rate at which data of the plurality of files is prefetched is controlled by pacing the prefetching based upon requests for data. This is very different from the teachings of Lynch and Sollars, either alone or in combination.

As explicitly admitted in the Office Action, Lynch does not disclose applicants' claimed feature of controlling, subsequent to determining that the prefetching of data is to occur, a rate at which the data is prefetched by pacing the prefetching based upon requests for data. Thus, Sollars is relied upon. However, Sollars does not overcome the deficiencies of Lynch.

While Sollars teaches pacing of prefetching, the prefetching of Sollars is of instructions and not of data. Sollars describes instruction prefetch in a micro-processor (e.g., Col. 4, lines 59-

67). This is in contrast to applicants' claimed invention in which the prefetching is of data and not of instructions. Further, the prefetching in Sollars is not paced based upon requests for data, but rather on how long it takes various parts of the hardware (pipelines) to process that data. Thus, Sollars fails to describe, teach or suggest pacing the prefetching of data based upon requests for data.

Since neither Lynch nor Sollars describes, teaches or suggests applicants' claimed element of controlling, subsequent to determining that prefetching of data is to occur, a rate at which data of a plurality of files is prefetched by pacing the prefetching based upon requests for data, applicants respectfully submit that the combination of Lynch and Sollars also fails to teach or suggest this claimed element. Thus, applicants respectfully requests an indication of allowability for independent claim 12, similar independent claims and all claims that depend therefrom.

In yet a further aspect of applicants' invention, applicants claim a method of managing the prefetching of inodes associated with files of a directory, the directory including one or more directory blocks and each directory block having associated therewith zero or more files (e.g., independent claim 17). The method includes, for instance, detecting a pattern of requests for multiple inodes associated with multiple files of a directory block of the one or more directory blocks, wherein the pattern is based on directory entries of the multiple files being within the directory block; and prefetching a plurality of inodes associated with the directory block, in response to detecting the pattern. Thus, in this aspect of applicants' claimed invention, the pattern that is detected is based upon directory entries of multiple files being within a directory block. This is very different from the teachings of Lynch and Mason, either alone or in combination.

For example, Lynch discloses a microprocessor configured to detect a group of instructions. There is no discussion in Lynch of detecting a pattern that is based on directory entries of multiple files being within a directory block. As a matter of fact, there is no description in Lynch of directories or directory blocks. Since directories or directory blocks are not even mentioned, it follows that there is no teaching or suggestion in Lynch of detecting a

pattern...wherein the pattern is based on directory entries of the multiple files being within the directory block.

Support for this rejection is indicated in the Office Action, in which it states, "The multiple files correspond to the files in which the data elements belong to," and "The directory block is comprised of all cache lines in the cache and thus, the detection of the pattern is based on all cache entries." Applicants respectfully submit that even if the data that is being prefetched belongs to one or more files, the processing unit in Lynch has no way of knowing this. Lynch describes a detection pattern that is not related to whether or not the data is associated with files, the directory, or a directory block. Instead, the pattern of data access is based on analyzing registers of instructions and the formation of memory addresses (e.g., an address being repeatedly incremented by a constant value to obtain other addresses; see Col. 9, lines 14-23). This is very different from a pattern based on directory entries of multiple files being with a directory block, as recited by one or more claims presented herewith.

Further, applicants respectfully disagree that a cache is equivalent to a directory block. As known in the art and as described in applicants' specification (see, e.g., page 8, lines 1-5 and FIG. 3), a directory block is a block of information in a directory. A directory is defined as an organizational unit or container used to organize files into a hierarchical structure (see, e.g., Webopedia Computer Dictionary at www.pc.webopedia.com/term/d/directory.html). In contrast to a directory, a cache is just a place to store data that does not have the hierarchical structure of a directory. Thus, applicants respectfully submit that a directory is much different than a cache. Since Lynch fails to describe, teach or suggest a directory, a directory block, or detecting a pattern that is based on directory entries of multiple files being within a directory block, applicants respectfully submit that their claimed invention is not taught or suggested by Lynch.

Mason fails to overcome the above described deficiencies of Lynch. Mason describes a disk drive array controller that allows prefetching of disk drive metadata used exclusively by the disk array controller and other disk subsystem components for the controller maintenance of the disk array system. The metadata in Mason includes parity information (e.g., parity blocks) (Col. 1, lines 41-47). The prefetching of metadata in Mason is quite different from the inode prefetch management technique of one aspect of applicants' claimed invention.

Like Lynch, Mason fails to even mention a directory much less applicants' claimed element of detecting a pattern in which the pattern is based on directory entries of the multiple files being within a directory block. In contrast, Mason's prefetch of metadata is based on the likelihood of one command following another, without regard to whether the metadata is associated with multiple files of a directory block. For example, since a write command (which involves parity blocks) often follows a read of a block, Mason prefetches a parity block into a cache to save time and enhance performance of input/output (I/O) processing (see col. 7, lines 54-63). This parity block reflects the internal structure of the storage system in Mason, and is not user-defined or visible to the user. Thus, this parity block prefetching in Mason is based on internal system structure, rather than on files being within the same directory, let alone on directory entries of those files being within a directory block, as recited by one or more claims presented herewith.

The Office Action cites col. 4, lines 30-41; col. 7, lines 53-63; and col. 9, lines 32-39 of Mason as teaching prefetching of metadata. These sections of Mason describe a prefetch technique that includes prefetching metadata (e.g., parity information/blocks) into a backend cache coordinated with a frontend cache, in parallel with other operations, and in conjunction with a write command because it is likely to follow a read of a block. Although prefetching of metadata is described in these sections of Mason, they do not teach or suggest prefetching in response to detecting the pattern described above. Again, it is the detection of the pattern based on the above-noted directory entries that is not taught or suggested by Mason.

Since both Lynch and Mason fail to teach or suggest applicants' claimed element of detecting a pattern of requests for multiple inodes associated with multiple files of a directory block, wherein the pattern is based on directory entries of the multiple files being within the directory block, applicants respectfully submit that the combination also fails to teach or suggest this claimed element. Therefore, applicants respectfully request an indication of allowability for independent claim 17, similar independent claims, and all claims that depend therefrom.

The dependent claims are patentable for the same reasons as the independent claims from which they directly or ultimately depend, as well as for their own additional features. For example, in dependent claim 3, applicants explicitly claim that the multiple files and the plurality

POU920000091US1

- 7 -

of files described above relative to claim 1 are within a single directory. That is, the multiple files whose data is requested in the detected pattern described above, and the plurality of files whose data is prefetched in response to detecting the pattern are located within the same directory.

In contrast, none of the applied art teaches or suggests that multiple files whose data is requested in a pattern and the plurality of files whose data is prefetched in response to detecting the pattern are located within a single directory. For example, as noted above, Lynch prefetches data according to the pattern established by detecting a group of instructions that access a next memory address determined by, for instance, adding a constant value to a current memory address. There is no teaching or suggestion in Lynch that the group of instructions request access to data in files of a directory at all, or that the prefetched data is associated with files of a directory, let alone that such files, if they exist, are located in a single directory, as recited by the present invention.

In the Office Action, it is stated that the multiple files and the plurality of files being associated with a single directory is disclosed in Lynch's system that inherently includes a page table/translation table for the main memory depicted in FIG. 5 thereof. Applicants respectfully traverse the alleged equivalency between applicants' recited directory and a page table/translation table. Applicants' recited directory reflects a structure of data in files that is user-defined and visible to the user. In contrast, a page table or translation table is a structure internal to Lynch's system that allows, for instance, the tracking of the location of data in memory at a given time. These internal table structures are not visible to the user, nor are they user-defined. Moreover, the definition of a directory is very different from a table as used in the art. Again, a directory is an organizational unit to organize files. In contrast, a table does not organize files but mainly holds data in a row, column format. One skilled in the art would not consider the structures equivalent. Thus, applicants respectfully submit that claim 3 and other similar dependent claims are patentable over Lynch, alone or in combination with the other applied patents.

Should the Examiner wish to discuss this case with applicants' attorney, please contact applicants' attorney at the below listed number.

Respectfully submitted,

Blanche E. Schiller
Blanche E. Schiller
Attorney for Applicants
Registration No.: 35,670

Dated: April 23, 2004.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579